# HANDLEIDING 'SYSTEM EDITOR' EN 'DOCUMENT PROCESSOR'

De Groep Janssen

Janssen en Janssen Automatisering B.V./Raadgevend Buro G.J.M. Janssen B.V.

Postbus 189, 3700 AD Zeist 03404-50520

# ALGEMEEN

De 'system editor' voor het systeem BASF-7100 vervaardigt en verwerkt tekstbestanden. Met behulp van deze 'system editor' kunt u brieven, dokumentatie, e.d. vervaardigen.

Het starten van de 'system editor' : 'run loadedit:0'.

De ingave van een tekstparagraaf is eenvoudig. Inplaats van een opdracht geeft u een paragraafnummer, de gewenste tekst, EOM en ENTER in. Paragraafnummers tussen 0 en 9999 zijn toegestaan. Een paragraaf wordt niet begrensd door een regel op het beeldscherm, deze kan zich over het gehele beeldscherm uitstrekken. Het verdient echter wel aanbeveling om enige regels op het beeldscherm niet te gebruiken, daar anders na het ingeven van ENTER het door u ingegeven paragraafnummer naar boven 'weggescrolled' wordt, waardoor uw opdracht dus ongeldig en uw ingave dus tevergeefs was.

Alle 'system editor'-bestanden zijn zogenaamde index sequentiele bestanden. Dit houdt in dat de volgorde van inbrengen van paragraafnummers volledig vrij is. Het paragraafnummer geeft namelijk de volgorde aan en met dit paragraafnummer kunt u ook rechtstreeks tekstparagrafen oproepen.

De hierna volgende EDITOR opdrachten zijn toegestaan:

3 COPYFILE........Kopieren van een bestand.
GET.............Openen en laden van een bestaand bestand.
9 PRINT...........Niet ge-edit printen van het bestand.
3 DELETE..........Verwijderen van een paragraaf.
5 ERASE...........Verwijderen van een bestand.
LIST............Lijsten van het bestand op het beeldscherm.
11 WHAT............Opvragen van de huidige bestandsnaam.
4 DOCU............Ge-edit printen van het bestand.
7 MAKE............Definieren en aanmaken van een nieuw bestand.
10 RENUM...........Hernummeren van het bestand.
6 FIND............Zoeken naar een bepaalde tekenkombinatie.
8 MERGE...........Samenvoegen van meerdere bestanden.
9 REORG...........Reorganiseren van het bestand.
11 STATUS..........Opvragen status van het bestand c.q. diskette.

Een gedetaillerde beschrijving van deze opdrachten volgt op de volgende pagina's.

## COPYFILE

Deze opdracht wordt gebruikt om een bestand te kopieren.
De syntax van deze opdracht is als volgt:

```
COPYFILE <file1:unit>,<file2:unit>
```

Voorbeeld:

```
COPYFILE Brief-1:1,Brief-2:1
```

## DELETE

Deze opdracht verwijdert reeds bestaande paragrafen.
De syntax van deze opdracht is als volgt:

```
DELETE <paragraafnummer>
```

Voorbeeld:

```
DELETE 500   (Verwijdert paragraaf met nummer 500)
```

Is de hier opgegeven paragraaf niet aanwezig, dan wordt dit met foutmelding 793 aangegeven.

## DOCU

Deze opdracht roept de 'document processor' aan. Deze zal een lijst aan de hand van de door u aangemaakte aanwijzingen en tekstregels samenstellen. Voor de verwerking wordt gebruik gemaakt van het bestand dat geopend is (d.w.z. aangeroepen m.b.v. een van de opdrachten GET of MAKE).
De syntax van deze opdracht is als volgt:

```
DOCU [<vanaf paragraaf> ,[<tot en met paragraaf>]]
```

De parameters in deze opdracht worden als volgt geinterpreteerd:

Wanneer er geen paragraafnummer opgegeven wordt, wordt het volledige bestand uitgeprint. Is er slechts 1 paragraafnummer opgegeven, dan wordt alleen de aangegeven paragraaf uitgeprint. Zijn er twee paragraafnummers opgegeven, dan worden alle paragrafen die tussen deze twee nummers liggen uitgeprint (inklusief de aangegeven paragrafen).

Wanneer de 'document processor' klaar is met het samenstellen van de lijst en het uitprinten daarvan, wordt automatisch de 'system editor' weer geladen, waarna u weer met de normale verwerking kunt verder gaan.

Een gedetailleerde beschrijving van de 'document processor' vindt u aan het einde van deze handleiding.

Voorbeeld:

```
DOCU 300,500  (vervaardigt een ge-editte print van de
               paragrafen 300 t/m 500).
```

## ERASE

Met behulp van deze opdracht kunnen bestanden verwijderd worden.

De syntax van de opdracht is als volgt:

ERASE <file:unit>

Voorbeeld:

ERASE bestand:1 (verwijdert het bestand 'bestand' op drive 1).

# FIND

De FIND opdracht vereenvoudigt het zoeken naar een bepaalde tekenkombinatie in het bestand. Het zoeken naar deze kombinatie geschiedt vanaf de huidige plaats in het bestand - tenzij anders wordt aangegeven.

De syntax van deze opdracht is als volgt:

```
FIND <scheidingsteken> <tekenkombinatie> <scheidingsteken>
                        [<paragraaf 1> [,<paragraaf 2>]]
```

Als scheidingsteken kunnen alle tekens (behalve de spatie) gebruikt worden. De scheidingstekens dienen echter wel zo gekozen te worden, dat ze zelf niet in de gezochte tekenkombinatie voorkomen.

Zijn er geen paragraafnummers opgegeven, dan wordt het bestand doorzocht vanaf de plaats waar u zich op het moment dat u deze opdracht geeft bevindt.

Indien de opgegeven tekenkombinatie niet gevonden wordt, dan staat de record-pointer op de laatste paragraaf die doorzocht is en wordt de volgende melding gegeven:

Text not found

Wordt er 1 paragraafnummer opgegeven dan begint het zoeken bij deze paragraaf en het zoeken stopt indien de kombinatie wordt gevonden of het einde van het bestand wordt bereikt. Worden er twee paragraafnummers opgegeven dan begint het zoeken bij de eerste paragraaf en stopt het zoeken indien de kombinatie wordt gevonden of dat ook de laatste paragraaf geheel is doorzocht. Wanneer de kombinatie is gevonden, dan wordt de paragraaf waarin deze voorkomt op het beeldscherm geplaatst.

Voorbeelden:

| | |
|---|---|
| FIND /nummer/ 100,200 | (doorzoekt alle paragrafen tussen 100 en 200 op 'nummer' en plaatst de paragraaf waarin deze kombinatie gevonden wordt op het beeldscherm. |
| FIND ?otto? 500 | (doorzoekt het bestand vanaf 500 op 'otto'. |

## MAKE

Deze opdracht maakt een nieuw bestand volgens de door de gebruiker opgegeven specifikaties aan.

De syntax van deze opdracht is als volgt:

MAKE <file:unit>,<max. aantal paragrafen>,T

De naam van de 'file' is vrij te kiezen, maar mag nog niet voorkomen op de diskette. Het maximaal aantal paragrafen mag niet worden overschreden (met behulp van de opdracht MERGE is het echter nog wel mogelijk dit maximum te wijzigen). De 'T' moet opgegeven worden, daar het bestand anders niet meer als tekstbestand te gebruiken is (dit geldt met name voor wat betreft de opdrachten DOCU en MERGE).

Voorbeeld:

MAKE beschr:1,200,T (Maakt een bestand aan op drive 1 onder de naam 'beschr', met als maximaal aantal paragrafen 200).

## MERGE

Met behulp van deze opdracht kunnen delen van verschillende bestanden toegevoegd worden aan het in bewerking zijnde bestand.

De syntax van deze opdracht is als volgt:

MERGE <file:unit>, [<paragraaf 1> [,<paragraaf 2>]]

Worden er geen paragraafnummers opgegeven, dan wordt het gehele bestand aan het in bewerking zijnde bestand toegevoegd.

WORDT ER EEN PARAGRAAFNUMMER GEVONDEN, DAT AL IN HET IN BEWERKING ZIJNDE BESTAND VOORKOMT, WORDT DEZE PARAGRAAF OVERSCHREVEN MET DE INHOUD VAN DE PARAGRAAF MET HET ZELFDE NUMMER UIT HET TOE TE VOEGEN BESTAND.

Wordt er een paragraafnummer opgegeven, dan wordt alleen die paragraaf aan het in bewerking zijnde bestand toegevoegd.

Worden er twee paragraafnummers opgegeven, dan worden alle paragrafen die tussen ~~tussen~~ deze twee paragraafnummers liggen aan het in bewerking zijnde bestand toegevoegd.

In al deze gevallen geldt dat het bestand van waaruit er toegevoegd word onveranderd blijft.

Voorbeelden:

| | |
|---|---|
| MERGE BESCHR:1 | (Voegt het gehele bestand 'BESCHR' toe aan het in bewerking zijnde bestand). |
| MERGE faktuur:1 10,30 | (Voegt de paragrafen 10 t/m 30 uit het bestand 'faktuur' toe aan het in bewerking zijnde bestand). |

## PRINT

Met behulp van deze opdracht kunt u het bestand uit printen in de vorm zoals u het heeft aangemaakt. Dit betekent, dat alle besturingstekens ten behoeve van de 'document processor' ook geprint worden.

De syntax van deze opdracht is als volgt:

```
PRINT [<paragraaf 1> [,<paragraaf 2>]]
```

De betekenis van de parameters is identiek aan die bij de opdracht DOCU.

## REORG

De bestanden aangemaakt met behulp van de 'system editor' zijn index-sequentieel opgebouwd, dit betekent, dat na tussenvoegen en/of verwijderen van paragrafen, gebieden op de diskette kunnen zijn ontstaan die niet met juiste gegevens zijn beschreven. Om deze gebieden vrij te geven en daarna weer ter beschikking te krijgen, moet er 'gereorganiseerd' worden. Dit gebeurt met behulp van de opdracht REORG.

De syntax van deze opdracht is als volgt:

```
REORG
```

Deze opdracht heeft geen parameters.

## RENUM

Met behulp van deze opdracht wordt het bestand opnieuw genummerd.

De syntax van deze opdracht is als volgt:

RENUM [<regel 1> [,<regel 2> [,<regel 3>]]]

Zonder parameters wordt met behulp van RENUM het gehele bestand opnieuw genummerd. De eerste paragraaf wordt dan 10 terwijl de daarop volgende paragrafen steeds een nummer krijgen dat telkens 10 hoger ligt.

KOMT TIJDENS ZO'N HERNUMMERING EEN PARAGRAAF OP EEN NUMMER UIT DAT HOGER LIGT DAN 9999 WORDT DE HERNUMMERING NIET UITGEVOERD EN VOLGT FOUTMELDING 205.

Indien er een parameter wordt opgegeven wordt eveneens het gehele bestand hernummerd, met dien verstande dat de eerste paragraaf het opgegeven regelnummer krijgt, terwijl de daarop volgende paragrafen steeds een nummer krijgen dat telkens 10 hoger ligt.

Indien er twee parameters worden opgegeven wordt eveneens het gehele bestand hernummerd, met dien verstande dat de eerste paragraaf het paragraafnummer krijgt dat als eerste parameter is opgegeven, terwijl de daarop volgende paragrafen steeds een nummer krijgen dat telkens met de waarde van de tweede parameter opgehoogd wordt.

De derde parameter geeft (indien opgegeven) aan met welk bestaand paragraafnummer de hernummering moet beginnen
DE DERDE PARAMETER MOET DAAROM ALTIJD KLEINER ZIJN DAN DE EERSTE, DAAR DE HERNUMMERING VOLGORDELIJK PLAATS VINDT (VAN 'ONDER' NAAR 'BOVEN').

Voorbeelden:

| | |
|---|---|
| RENUM | (Hernummert het gehele bestand. Eerste paragraaf wordt 10 ophoging daarna is ook 10). |
| RENUM 500 | (Als hierboven, de eerste paragraaf in dit geval wordt echter 500). |
| RENUM 50,25 | (Eerste paragraafnummer wordt 50, ophogingsfaktor 25). |
| RENUM 100,15,35 | (Eerste nieuwe paragraafnummer wordt 500, ophogingsfaktor 15 terwijl als eerste paragraaf 35 wordt hernummerd). |

## STATUS

Met behulp van deze opdracht kan de status van een bestand of de inhoud van een gehele diskette worden opgevraagd.

De syntax van deze opdracht is als volgt:

```
STATUS <[file]:unit>
```

Voorbeelden:

```
STATUS BEST:1   (geeft de status van het bestand 'BEST').

STATUS :1       (geeft de namen van alle bestanden die
                 zich op de diskette in unit 1 bevinden).
```

## WHAT

Met behulp van deze opdracht wordt de status van het in bewerking zijnde bestand opgevraagd.

De syntax van deze opdracht is als volgt:

```
WHAT
```

De opdracht heeft geen parameters.

## FOUTMELDINGEN

De volgende fouten kunnen tijdens de werking van de 'system editor' optreden:

COMMAND ERRORS (opdracht - fouten)
DATA ERRORS (ingave - fouten)
OVERLAY ERRORS (overlay - fouten)

Tevens is het mogelijk dat fouten uit de 700-serie optreden, de betekenis van deze fouten staat vermeld in de bijlage.

## TYPE 100 - COMMAND ERRORS

| | |
|---|---|
| 101 | Onjuiste bestandsnaam |
| 102 | Onjuiste opdracht |
| 103 | Geen geopend bestand aanwezig |
| 104 | Foutief paragraafnummer |
| 105 | Foutieve bestandsgrootte |
| 106 | Onjuist bestandstype |
| 107 | Foutieve parameter in RENUM opdracht |
| 108 | Niet nummeriek paragraafnummer |

## TYPE 200 - DATA ERRORS

| | |
|---|---|
| 201 | Onbekende tekst in assembler bestand |
| 202 | Onjuist assembler label |
| 203 | Foutieve trace aanroep |
| 204 | Bestandstypen ongelijk |
| 205 | Tijdens RENUM wordt een paragraafnummer groter dan 9999 |
| 206 | Onjuist soortnummer bij het printen |
| 207 | Foutieve tekenkombinatie in FIND opdracht |

## TYPE 300 - OVERLAY ERRORS

| | |
|---|---|
| 301 | Onjuist paragraafnummer in DOCU opdracht |
| 302 | Onjuiste tabulator, paginalengte en/of paginabreedte gezet in het bestand. |

## ALGEMEEN

De 'document processor' print een, met behulp van de 'system editor' aangemaakt, tekstbestand uit. Hierbij wordt gebruik gemaakt van de in dat bestand gebruikte besturingstekens.

De verwerking van een tekstbestand gebeurt als volgt:

Het bestand wordt gelezen en telkens wordt er gekeken of er in de eerste positie van een regel een stuurteken staat. Is dit niet het geval, dan wordt de betreffende regel geprint in de 'MODE' waarin de 'document processor' zich op dat moment bevindt.

Belangrijk zijn de INDELINGS- en PAGINA-MODE. Bevindt de 'document processor' zich in de 'indelings-mode' dan worden er in de af te drukken regel zoveel spaties tussen gevoegd als er nodig zijn om alle regels op de zelfde positie te laten beginnen en eindigen. Wordt er niet in deze 'mode' gewerkt, dan worden de regels geprint op de manier zoals ze zijn ingebracht. De 'pagina-mode' bewerkstelligt een automatische bladsprong wanneer het aantal opgegeven regels per pagina bereikt is. Tevens zorgt de 'pagina-mode' voor het printen van de opgegeven kopregels en een paginanummering.

De volgende besturingstekens zijn toegestaan:

<....DEFINIEERT DE EERSTE KOPREGEL.
<<...DEFINIEERT DE TWEEDE KOPREGEL.
@....DEFINIEERT EEN TEKENKOMBINATIE DIE IN DE PAGINANUMMERING OPGENOMEN MOET WORDEN.
&....BLADSPRONG (ALTIJD).
?....BLADSPRONG (ONDER VOORWAARDEN).
*....REGELSPRONG.
+....'INDELINGS-MODE' AANZETTEN.
-....'INDELINGS-MODE' UITZETTEN.
>....GEEFT PAGINA NUMMER AAN.
%....GEEFT 1e PRINT POSITIE AAN.
$....GEEFT AANTAL REGELS PER PAGINA AAN.
#....GEEFT AANTAL POSITIES PER REGEL AAN.
~....GEEFT AANTAL SPATIES BIJ EEN NIEUWE ALINEA AAN.
)....PLAATST IN HET MIDDEN VAN DE REGEL.

## '<' DEFINEREN VAN DE EERSTE KOPREGEL

Het teken '<' laat de 'document processor' weten, dat de hierachter aanwezige tekst na iedere bladsprong als eerste regel moet worden afgedrukt. Aan het einde van deze eerste regel wordt altijd het nieuwe paginanummer afgedrukt. (Deze wordt automatisch bij elke nieuwe pagina opgehoogd).

Is er eerder in het bestand al een eerste kopregel gedefineerd, dan wordt deze overschreven.

Volgt er op dit stuurteken geen tekst, dan wordt de eerste kopregel noch het paginanummer afgedrukt.

## '<<' DEFINEREN VAN DE TWEEDE KOPREGEL

Het teken '<<' laat de 'document processor' weten, dat de hierachter aanwezige tekst na iedere bladsprong als tweede regel moet worden afgedrukt. Ontbreekt de eerste kopregel dan wordt alleen deze kopregel afgedrukt (zonder paginanummering). Is er dus een kopregel nodig, zonder dat er een paginanummering gewenst is dan kunt u dat met behulp van dit stuurteken bereiken.

Voorbeeld:

<<DEEL 1      (plaatst als kopregel de tekst 'DEEL 1').

## '@' TEKENKOMBINATIE IN PAGINANUMMER

Het teken '@' laat de 'document processor' weten, dat de hierachter aanwezige tekst na iedere bladsprong in het paginanummer moet worden opgenomen. (Eis: de eerste kopregel (stuurteken '<') moet eerder zijn gedefinieerd).

Voorbeeld:

@IV-      (zet op bijv. pagina 3      PAGE IV-3).
@         (verwijdert de bestaande gegevens).

## '&' BLADSPRONG (ALTIJD)

Het teken '&' laat de 'document processor' weten, dat er onmiddelijk naar een nieuwe pagina moet worden gegaan. Eventueel gedefineerde kopregels en daarna twee lege regels worden afgedrukt.

## '?' BLADSPRONG (ONDER VOORWAARDEN)

Het teken '?' laat de 'document processor' weten, dat er, indien het achter dit teken opgegeven aantal plus het aantal al afgedrukte regels groter is dan het aantal regels per pagina, onmiddelijk naar een nieuwe pagina moet worden gegaan. Eventueel gedefineerde kopregels en daarna twee lege regels worden afgedrukt.

Voorbeeld:

```
        We nemen aan dat er 60 regels per pagina geprint worden.
?15     (huidig regelnummer 48 --> bladsprong
         huidig regelnummer 44 --> geen bladsprong).
```

## '*' REGELSPRONG

Het teken '*' laat de 'document processor' weten, dat er een lege regel moet worden afgedrukt.

## '+' 'INDELINGS-MODE' AAN

Het teken '+' laat de 'document processor' weten, dat er het laatste teken van elke af te drukken regel op een vaste positie moet komen. Hierdoor wordt bereikt dat er ook aan de rechterzijde van het dokument een rechte kantlijn ontstaat. Voorzover niet anders gedefinieerd (door #) wordt iedere regel door het invoegen van spaties op 72 posities gebracht.

## '-' 'INDELINGS-MODE' UIT

Het teken '-' laat de 'document processor' weten, dat indien de 'indelings-mode' aan is gezet met behulp van het teken '+' deze weer uitgezet moet worden. Alle regels die nu volgen worden geprint zoals ze op het beeldscherm zijn ingegeven.

## '>' ZET PAGINANUMMER

Het teken '>' laat de 'document processor' weten, dat er op de volgende pagina een ander paginanummer moet komen als dat er berekend is.

Voorbeelden:

>15 (volgende pagina krijgt paginanummer 15)

>1 (volgende pagina krijgt paginanummer 1)

## '$' ZET REGELS PER PAGINA

Het teken '$' laat de 'document processor' weten, hoeveel regels er op een pagina afgedrukt moeten worden.

## '#' ZET TEKENS PER REGEL

Het teken '#' laat de 'document processor' weten, hoeveel tekens er op een regel moeten worden afgedrukt. Dit stuurteken heeft echter alleen dan een funktie wanneer het in kombinatie met het teken '+' wordt gebruikt.

## '~' ZET AANTAL SPATIES BIJ NIEUWE ALINEA

Het teken '~' laat de 'document processor' weten, hoeveel spaties er aan het begin van een alinea aan het begin van de eerste regel moeten worden afgedrukt.

## ')' ZET IN HET MIDDEN VAN DE REGEL

Het teken ')' laat de 'document processor' weten, dat de achter dit teken aanwezige tekst in het midden van de regel moeten worden afgedrukt. Dit kan bijvoorbeeld gebruikt worden bij de hoofdstuknaam.

Voorbeeld:

```
)INHOUDSOPGAVE
)=============
```

# APPENDIX

Als uitgebreid voorbeeld van een met de 'system editor' opgebouwd en met de 'document processor' geprint tekstbestand, vindt u achter deze handleiding een print van deze beschrijving, inklusief alle stuurtekens.

Let u hierbij op de gebruikte technieken:

1. Er mag geen tekst rechtstreeks achter het paragraafnummer worden ingegeven.

2. De eerste positie van elke regel wordt uitsluitend gebruikt voor stuurtekens. Dit is echter niet verplicht, maar verhoogt wel de leesbaarheid van het tekstbestand tijdens eventuele korrekties.

## A3.6 FILE SUB SYSTEM ERRORS (700-799).

ERROR

701 A READ variable list is longer than the number of data items.

703 Unusable BASIC file. The designated file is not a BASIC program file, or some error has been encountered when trying to GET, RUN or CHAIN the file.

706 File name error on SAVE, RUN or chain. File name limited to 16 characters followed by a colon and unit number.

709 Disk not ready to load BASIC file.

710 Checksum error. System code loaded from diskette has failed to meet a checksum test. Try loading from another diskette.

732 Read after write failure. On a read check after a write operation the data did not compare.

733 Busy drive. A unit appears to be busy when no operation has been issued.

734 Lost data. During a read or write operation data was lost.

736 Same as 734.

740 CRC error. During a read operation the CRC byte did not match the data.

748 Abort. A copy diskette, or initialize operation was aborted by use of the break button. The operation is incomplete.

749 No file. This error will occur if an attempt is made to open a file which is not present on the specified device, or to read/write a file not previously OPENed.

750 Protection error. This error will occur if an attempt is made to access a protected file.

751 Duplicate file. An attempt has been made to create a file which is already present on the specified drive.

752 Bad Operation. The last operator character is invalid.

CONTINUATION

ERROR

753 Bad System. An irrecoverable error in the disk system has been detected.

754 Long string. The length of the string or fields being written to the file exceed the record size of the file.

755 Disk overflow. There is not enough room on the diskette to store another page of the file, or directory is full.

756 Bounds error. An attempt has been made to read, write, or positon a file after the last record or before the first record of the file. Use the function STATUS to determine the actual length of the file.

757 Write protect. An attempt has been made to write to a file which has been write protected (see ATTRIB function) or to a diskette which has the write protect tab on. This error can also occur if an attempt is made to change the name of a file, change the name of a diskette, change the user label on a diskette, initialize a diskette or update a file when the diskette specified is not write ready.
a file, change the name of a diskette, initialize a diskette or update a file when the diskette specified is not write ready.

758 Disk locked.

759 File locked.

765 Bad Sector. The result of a read or write operation has returned an error status.

766 Bad parameter/file name. A parameter to a disk command is invalid, or a file/diskettte name contains invalid characters.

768 Bad Seek. The last attempted seek has failed to located the desired track on diskette after repeated tries.

769 Null Buffer. In an exchange buffers operation the disk has 'lost' the buffer. The exchange buffers operation is used by the BASIC interpreter to speed up overlay operations.

CONTINUATION

ERROR

771 Ram error. During self check at cold start the ram test failed.

772 Allocaton Error. The file allocation and the allocation according to the AST do not agree. All files should be copied to another disk as system performance from this point is questionable.

773 Too many open files. There is not enough space in the disk subsystem RAM to accommodate another file. Close files which are no longer needed.

774 Update error. During a directory update operation certain critical fields do not match the FIB contained on diskette. This may arise from changing diskettes without closing files refering to this diskette.

775 Unusable diskette. During an initialization operation more than 20 errors have occurred, or the system sectors could not be written/read after initialization. Discard the diskette. If the problem persists it may be a drive malfunction.

777 Timeout. An operation on the disk drive has not finished in sufficient time.

780 Prom error. This error may occurs when the disk system is first turned on, or whenever the disk system is cleared. It indicates that the sum check of the system firmware failed. This usually indicates a faulty PROM. However, if your system has been recently updated it may indicate that a PROM has been inserted incorrectly.

781 Sector I.D. or data Address mark not found during a disk operation.

782 Bad data address mark found during disk operation.

783 Bad data address mark found during disk operation.

784 Bad data address mark found during disk operation.

786 Bad Key. A key to an indexed sequential file contains unacceptable characters (zero or greater than @F0).

788 Not ready. Drive not ready to accept a command. Be sure that desired diskette is properly inserted.

789 RAM error. This error may occur when the disk system is first turned on, or whenever the disk system is cleared. It indicates a bad RAM in the system.

793 Key not found in an indexed sequential file.

```
PROGRAM EDITDOCU:2

0010
<HANDLEIDING 'SYSTEM EDITOR' & 'DOCUMENT PROCESSOR'
<<'SYSTEM EDITOR'
#65
~3
%5
>1
$65

0020
&
)ALGEMEEN
)========
*
+De 'system editor' voor het systeem BASF-7100 vervaardigt en verwerkt
 tekstbestanden. Met behulp van deze 'system editor' kunt u brieven,
 dokumentatie, e.d. vervaardigen.
*Het starten van de 'system editor' : 'run loadedit:0'.

0030
*De ingave van een tekstparagraaf is eenvoudig. Inplaats van een opdracht
 geeft u een paragraafnummer, de gewenste tekst, EOM en ENTER in.
 Paragraafnummers
 tussen 0 en 9999 zijn toegestaan. Een paragraaf wordt niet begrensd door
 een regel op het beeldscherm, deze kan zich over het gehele beeldscherm
 uitstrekken. Het verdient echter wel aanbeveling om enige regels op het
 beeldscherm niet te gebruiken, daar anders na het ingeven van ENTER het
 door u ingegeven paragraafnummer naar boven 'weggescrolled' wordt, waardoor
 uw opdracht dus ongeldig en uw ingave dus tevergeefs was.

0040
*Alle 'system editor'-bestanden zijn zogenaamde index sequentiele bestanden.
 Dit houdt in dat de volgorde van inbrengen van paragraafnummers volledig vrij
 is. Het paragraafnummer geeft namelijk de volgorde aan en met dit
 paragraafnummer kunt u ook rechtstreeks tekstparagrafen oproepen.

0050
&
*
-De hierna volgende EDITOR opdrachten zijn toegestaan:
*COPYFILE........Kopieren van een bestand.
 GET.............Openen en laden van een bestaand bestand.
 PRINT...........Niet ge-edit printen van het bestand.
 DELETE..........Verwijderen van een paragraaf.
 ERASE...........Verwijderen van een bestand.
 LIST............Lijsten van het bestand op het beeldscherm.
 WHAT............Opvragen van de huidige bestandsnaam.
 DOCU............Ge-edit printen van het bestand.
 MAKE............Definieren en aanmaken van een nieuw bestand.
 RENUM...........Hernummeren van het bestand.
 FIND............Zoeken naar een bepaalde tekenkombinatie.
 MERGE...........Samenvoegen van meerdere bestanden.
 REORG...........Reorganiseren van het bestand.
 STATUS..........Opvragen status van het bestand c.q. diskette.
```

```
0060
*
+Een gedetaillerde beschrijving van deze opdrachten volgt op de volgende
 pagina's.

0070
&
)COPYFILE
)--------
*
-Deze opdracht wordt gebruikt om een bestand te kopieren.
 De syntax van deze opdracht is als volgt:
*   COPYFILE <file1:unit>,<file2:unit>
*Voorbeeld:
*   COPYFILE Brief-1:1,Brief-2:1

0080
*
*
)DELETE
)------
*
-Deze opdracht verwijdert reeds bestaande paragrafen.
 De syntax van deze opdracht is als volgt:
*   DELETE <paragraafnummer>
*Voorbeeld:
*   DELETE 500   (Verwijdert paragraaf met nummer 500)
*
+Is de hier opgegeven paragraaf niet aanwezig, dan wordt dit met foutmelding
 793 aangegeven.

0090
&
)DOCU
)----
*
+Deze opdracht roept de 'document processor' aan. Deze zal een lijst
 aan de hand van de door u aangemaakte aanwijzingen en tekstregels
 samenstellen. Voor de verwerking wordt gebruik gemaakt van het bestand
 dat geopend is (d.w.z. aangeroepen m.b.v. een van de opdrachten GET of
 MAKE).
-De syntax van deze opdracht is als volgt:
*   DOCU [<vanaf paragraaf> ,[<tot en met paragraaf>]]
*De parameters in deze opdracht worden als volgt geinterpreteerd:
```

```
0100
*
+Wanneer er geen paragraafnummer opgegeven wordt, wordt het volledige
 bestand uitgeprint. Is er slechts 1 paragraafnummer opgegeven, dan wordt
 alleen de aangegeven paragraaf uitgeprint. Zijn er twee paragraafnummers
 opgegeven, dan worden alle paragrafen die tussen deze twee nummers liggen
 uitgeprint (inklusief de aangegeven paragrafen).
*Wanneer de 'document processor' klaar is met het samenstellen van de
 lijst en het uitprinten daarvan, wordt automatisch de 'system editor'
 weer geladen, waarna u weer met de normale verwerking kunt verder gaan.
*Een gedetailleerde beschrijving van de 'document processor' vindt u
 aan het einde van deze handleiding.
-
 Voorbeeld:
*   DOCU 300,500  (vervaardigt een ge-editte print van de
                   paragrafen 300 t/m 500).

0110
&
)ERASE
)-----
*
+Met behulp van deze opdracht kunnen bestanden verwijderd worden.
-
 De syntax van de opdracht is als volgt:
*   ERASE <file:unit>
*Voorbeeld:
*   ERASE bestand:1  (verwijdert het bestand 'bestand' op drive 1).

0120
&
)FIND
)----
*
+De FIND opdracht vereenvoudigt het zoeken naar een bepaalde tekenkombinatie
 in het bestand. Het zoeken naar deze kombinatie geschiedt vanaf de huidige
 plaats in het bestand - tenzij anders wordt aangegeven.
-
 De syntax van deze opdracht is als volgt:
*   FIND <scheidingsteken> <tekenkombinatie> <scheidingsteken>
                              [<paragraaf 1> [,<paragraaf 2>]]
*
+Als scheidingsteken kunnen alle tekens (behalve de spatie) gebruikt worden.
 De scheidingstekens dienen echter wel zo gekozen te worden, dat ze zelf
 niet is de gezochte tekenkombinatie voorkomen.
```

```
0130
*Zijn er geen paragraafnummers opgegeven, dan wordt het bestand doorzocht
 vanaf de plaats waar u zich op het moment dat u deze opdracht geeft
 bevind.
*Indien de opgegeven tekenkombinatie niet gevonden wordt, dan staat de
 record-pointer op de laatste paragraaf die doorzocht is en wordt de
 volgende melding gegeven:
*Text not found
*Wordt er 1 paragraafnummer opgegeven dan begint het zoeken bij deze
 paragraaf en het zoeken stopt indien de kombinatie wordt gevonden of
 het einde van het bestand wordt bereikt. Worden er twee paragraafnummers
 opgegeven dan begint het zoeken bij de eerste paragraaf en stopt het
 zoeken indien de kombinatie wordt gevonden of dat ook de laatste paragraaf
 geheel is doorzocht. Wanneer de kombinatie is gevonden, dan wordt de
 paragraaf waarin deze voorkomt op het beeldscherm geplaatst.

0140
-
 Voorbeelden:
*   FIND /nummer/ 100,200   (doorzoekt alle paragrafen tussen
                             100 en 200 op 'nummer' en plaatst
                             de paragraaf waarin deze kombinatie
                             gevonden wordt op het beeldscherm.
*   FIND ?otto? 500         (doorzoekt het bestand vanaf 500
                             op 'otto'.

0150
&
)MAKE
)----
*
+Deze opdracht
 maakt een nieuw bestand volgens de door de
 gebruiker opgegeven specifikaties aan.
*
-De syntax van deze opdracht is als volgt:
*   MAKE <file:unit>,<max. aantal paragrafen>,T
*
```

```
0160
+De naam van de 'file' is vrij te kiezen, maar mag nog niet voorkomen op
 de diskette. Het
 maximaal aantal paragrafen mag niet worden overschreden (met behulp van
 de opdracht MERGE is het echter nog wel mogelijk dit maximum te
 wijzigen). De 'T' moet opgegeven worden, daar het bestand anders niet meer
 als tekstbestand te gebruiken is (dit geldt met name voor wat betreft
 de opdrachten DOCU en MERGE).
-
 Voorbeeld:
*   MAKE beschr:1,200,T  (Maakt een bestand aan op drive 1 onder
                          de naam 'beschr', met als maximaal
                          aantal paragrafen 200).

0170
&
*
)MERGE
)-----
*
+Met behulp van deze opdracht kunnen delen van verschillende bestanden
 toegevoegd worden aan het in bewerking zijnde bestand.
-
 De syntax van deze opdracht is als volgt:
*   MERGE <file:unit>, [<paragraaf 1> [,<paragraaf 2>]]

+Worden er geen paragraafnummers opgegeven, dan wordt het gehele bestand
 aan het in bewerking zijnde bestand toegevoegd.
*WORDT ER EEN PARAGRAAFNUMMER GEVONDEN, DAT AL IN HET IN BEWERKING ZIJNDE
 BESTAND VOORKOMT, WORDT DEZE PARAGRAAF OVERSCHREVEN MET DE INHOUD VAN DE
 PARAGRAAF MET HET ZELFDE NUMMER UIT HET TOE TE VOEGEN BESTAND.
*Wordt er een paragraafnummer opgegeven, dan wordt alleen die paragraaf aan
 het in bewerking zijnde bestand toegevoegd.
*Worden er twee paragraafnummers opgegeven, dan worden alle paragrafen die
 tussen tussen deze twee paragraafnummers liggen aan het in bewerking zijnde
 bestand toegevoegd.

0180
*In al deze gevallen geldt dat het bestand van waaruit er toegevoegd word
 onveranderd blijft.
-
 Voorbeelden:
*   MERGE BESCHR:1          (Voegt het gehele bestand 'BESCHR' toe
                             aan het in bewerking zijnde bestand).
*   MERGE faktuur:1,10,30   (Voegt de paragrafen 10 t/m 30 uit het
                             bestand 'faktuur' toe aan het in
                             bewerking zijnde bestand).
```

0190
&
)PRINT
)-----
*
+Met behulp van deze opdracht kunt u het bestand uit printen in de vorm
zoals u het heeft aangemaakt. Dit betekent, dat alle besturingstekens
ten behoeve van de 'document processor' ook geprint worden.
-
De syntax van deze opdracht is als volgt:
* PRINT [<paragraaf 1> [,<paragraaf 2>]]
*
+De betekenis van de parameters is identiek aan die bij de opdracht DOCU.

0200
*
*
)REORG
)-----
*
+De bestanden aangemaakt met behulp van de 'system editor' zijn
index-sequentieel opgebouwd, dit betekent, dat na tussenvoegen en/of
verwijderen van paragrafen, gebieden op de diskette kunnen zijn ontstaan
die niet met juiste gegevens zijn beschreven. Om deze gebieden vrij te
geven en daarna weer ter beschikking te krijgen, moet er 'gereorganiseerd'
worden. Dit gebeurt met behulp van de opdracht REORG.
-
De syntax van deze opdracht is als volgt:
* REORG
*Deze opdracht heeft geen parameters.

0210
&
)RENUM
)-----
*
+Met behulp van deze opdracht wordt het bestand opnieuw genummerd.
-
De syntax van deze opdracht is als volgt:
* RENUM [<regel 1> [,<regel 2> [,<regel 3>]]]
*
+Zonder parameters wordt met behulp van RENUM het gehele bestand opnieuw
genummerd. De eerste paragraaf wordt dan 10 terwijl de daarop volgende
paragrafen steeds een nummer krijgen dat telkens 10 hoger ligt.
*KOMT TIJDENS ZO'N HERNUMMERING EEN PARAGRAAF OP EEN NUMMER UIT DAT
HOGER LIGT DAN 9999 WORDT DE HERNUMMERING NIET UITGEVOERD EN VOLGT
FOUTMELDING 205.

```
0220
*Indien er een parameter wordt opgegeven wordt eveneens het gehele bestand
 hernummerd, met dien verstande dat de eerste paragraaf het opgegeven
 regelnummer krijgt, terwijl de daarop volgende paragrafen steeds een nummer
 krijgen dat telkens 10 hoger ligt.
*Indien er twee parameters worden opgegeven wordt eveneens het gehele bestand
 hernummerd, met dien verstande dat de eerste paragraaf het paragraafnummer
 krijgt dat als eerste parameter is opgegeven, terwijl de daarop volgende
 paragrafen steeds een nummer krijgen dat telkens met de waarde van de
 tweede parameter opgehoogd wordt.
*De derde parameter geeft (indien opgegeven) aan met welk bestaand
 paragraafnummer de hernummering moet beginnen
+DE DERDE PARAMETER MOET DAAROM ALTIJD KLEINER ZIJN DAN DE EERSTE, DAAR
 DE HERNUMMERING VOLGORDELIJK PLAATS VIND (VAN 'ONDER' NAAR 'BOVEN').

0230
-
 Voorbeelden:
*   RENUM               (Hernummert het gehele bestand. Eerste
                         paragraaf wordt 10 ophoging daarna is
                         ook 10).
*   RENUM 500           (Als hierboven, de eerste paragraaf in
                         dit geval wordt echter 500).
*   RENUM 50,25         (Eerste paragraafnummer wordt 50,
                         ophogingsfaktor 25).
*   RENUM 100,15,35     (Eerste nieuwe paragraafnummer wordt 500,
                         ophogingsfaktor 15 terwijl als eerste
                         paragraaf 35 wordt hernummerd).

0240
&
)STATUS
)------
*
+Met behulp van deze opdracht kan de status van een bestand of de inhoud
 van een gehele diskette worden opgevraagd.
-
 De syntax van deze opdracht is als volgt:
*   STATUS <[file]:unit>
*Voorbeelden:
*   STATUS BEST:1   (geeft de status van het bestand 'BEST').
*   STATUS :1       (geeft de namen van alle bestanden die
                     zich op de diskette in unit 1 bevinden).
```

```
0250
*
*
)WHAT
)----
*
+Met behulp van deze opdracht wordt de status van het in bewerking zijnde
 bestand opgevraagd.
-
 De syntax van deze opdracht is als volgt:
*   WHAT
*De opdracht heeft geen parameters.

0260
&
)FOUTMELDINGEN
)-------------
*
+De volgende fouten kunnen tijdens de werking van de 'system editor' optreden:
-
)           COMMAND ERRORS (opdracht - fouten)
)           DATA ERRORS    (ingave   - fouten)
)           OVERLAY ERRORS (overlay  - fouten)
*
+Tevens is het mogelijk dat fouten uit de 700-serie optreden,
 de betekenis van deze fouten staat vermeld in de bijlage.

0270
*
)TYPE 100 - COMMAND ERRORS
)-------------------------
*
-      101    Onjuiste bestandsnaam
       102    Onjuiste opdracht
       103    Geen geopend bestand aanwezig
       104    Foutief paragraafnummer
       105    Foutieve bestandsgrootte
       106    Onjuist bestandstype
       107    Foutieve parameter in RENUM opdracht
       108    Niet nummeriek paragraafnummer
```

```
0280
*
)TYPE 200 - DATA ERRORS
)----------------------
*
-      201    Onbekende tekst in assembler bestand
       202    Onjuist assembler label
       203    Foutieve trace aanroep
       204    Bestandstypen ongelijk
       205    Tijdens RENUM wordt een paragraafnummer groter
              dan 9999
       206    Onjuist soortnummer bij het printen
       207    Foutieve tekenkombinatie in FIND opdracht

0290
*
)TYPE 300 - OVERLAY ERRORS
)-------------------------
*
-      301    Onjuist paragraafnummer in DOCU opdracht
       302    Onjuiste tabulator, paginalengte en/of
              paginabreedte gezet in het bestand.

0300
<<'DOCUMENT PROCESSOR'

0310
&
*
)ALGEMEEN
)========
*
+De 'document processor' print een, met behulp van de 'system editor'
 aangemaakt, tekstbestand uit. Hierbij wordt gebruik gemaakt van de in dat
 bestand gebruikte besturingstekens.
*De verwerking van een tekstbestand gebeurt als volgt:
*Het bestand wordt gelezen en telkens wordt er gekeken of er in de
 eerste positie van een regel een stuurteken staat. Is dit niet het geval,
 dan wordt de betreffende regel geprint in de 'MODE' waarin de
 'document processor' zich op dat moment bevindt.
```

0320
*Belangrijk zijn de INDELINGS- en PAGINA-MODE. Bevindt de 'document processor' zich in de 'indelings-mode' dan worden er in de af te drukken regel zoveel spaties tussen gevoegd als er nodig zijn om alle regels op de zelfde positie te laten beginnen en eindigen. Wordt er niet in deze 'mode' gewerkt, dan worden de regels geprint op de manier zoals ze zijn ingebracht. De 'pagina-mode' bewerkstelligt een automatische bladsprong wanneer het aantal opgegeven regels per pagina bereikt is. Tevens zorgt de 'pagina-mode' voor het printen van de opgegeven kopregels en een paginanummering.

0330
-
De volgende besturingstekens zijn toegestaan:

<....DEFINIEERT DE EERSTE KOPREGEL.
<<...DEFINIEERT DE TWEEDE KOPREGEL.
@....DEFINIEERT EEN TEKENKOMBINATIE DIE IN DE PAGINANUMMERING OPGENOMEN MOET WORDEN.
&....BLADSPRONG (ALTIJD).
?....BLADSPRONG (ONDER VOORWAARDEN).
*....REGELSPRONG.
+....'INDELINGS-MODE' AANZETTEN.
-....'INDELINGS-MODE' UITZETTEN.
>....GEEFT PAGINA NUMMER AAN.
%....GEEFT 1e PRINT POSITIE AAN.
$....GEEFT AANTAL REGELS PER PAGINA AAN.
#....GEEFT AANTAL POSITIES PER REGEL AAN.
~....GEEFT AANTAL SPATIES BIJ EEN NIEUWE ALINEA AAN.
)....PLAATST IN HET MIDDEN VAN DE REGEL.

0340
&
)'<' DEFINEREN VAN DE EERSTE KOPREGEL
)------------------------------------
*
+Het teken '<' laat de 'document processor' weten, dat de hierachter aanwezige tekst na iedere bladsprong als eerste regel moet worden afgedrukt. Aan het einde van deze eerste regel wordt altijd het nieuwe paginanummer afgedrukt. (Deze wordt automatisch bij elke nieuwe pagina opgehoogd).
*Is er eerder in het bestand al een eerste kopregel gedefineerd, dan wordt deze overschreven.
*Volgt er op dit stuurteken geen tekst, dan wordt de eerste kopregel noch het paginanummer afgedrukt.

```
0350
*
*
)'<<'  DEFINEREN VAN DE TWEEDE KOPREGEL
)---------------------------------------
*
+Het teken '<<' laat de 'document processor' weten, dat de hierachter
 aanwezige tekst na iedere bladsprong als tweede regel moet worden
 afgedrukt. Ontbreekt de eerste kopregel dan wordt alleen deze kopregel
 afgedrukt (zonder paginanummering). Is er dus een kopregel nodig, zonder
 dat er een paginanummering gewenst is dan kunt u dat met behulp van dit
 stuurteken bereiken.
-
 Voorbeeld:
*<<DEEL 1       (plaatst als kopregel de tekst 'DEEL 1').

0360
*
*
)'@'  TEKENKOMBINATIE IN PAGINANUMMER
)---------------------------------------
*
+Het teken '@' laat de 'document processor' weten, dat de hierachter
 aanwezige tekst na iedere bladsprong in het paginanummer moet worden
 opgenomen. (Eis: de eerste kopregel (stuurteken '<') moet eerder zijn
 gedefinieerd).
-
 Voorbeeld:
*@IV-        (zet op bijv. pagina 3      PAGE IV-3).
 @           (verwijdert de bestaande gegevens).

0370
&
)'&'  BLADSPRONG (ALTIJD)
)------------------------
*
+Het teken '&' laat de 'document processor' weten, dat er onmiddelijk naar
 een nieuwe pagina moet worden gegaan. Eventueel gedefineerde kopregels
 en daarna twee lege regels worden afgedrukt.
```

```
0380
*
*
)'?' BLADSPRONG (ONDER VOORWAARDEN)
)-----------------------------------
*
+Het teken '?' laat de 'document processor' weten, dat er, indien het achter
 dit teken opgegeven aantal plus het aantal al afgedrukte regels groter is
 dan het aantal regels per pagina, onmiddelijk naar
 een nieuwe pagina moet worden gegaan. Eventueel gedefineerde kopregels
 en daarna twee lege regels worden afgedrukt.
-
 Voorbeeld:
*       We nemen aan dat er 60 regels per pagina geprint worden.
 ?15    (huidig regelnummer 48 --> bladsprong
         huidig regelnummer 44 --> geen bladsprong).

0390
*
*
)'*'  REGELSPRONG
)---------------
*
+Het teken '*' laat de 'document processor' weten, dat er een lege regel moet
 worden afgedrukt.

0400
*
*
)'+'  'INDELINGS-MODE' AAN
)-----------------------
*
+Het teken '+' laat de 'document processor' weten, dat er het laatste teken va
 elke af te drukken regel op een vaste positie moet komen. Hierdoor wordt
 bereikt dat er ook aan de rechterzijde van het dokument een rechte kantlijn
 ontstaat. Voorzover niet anders gedefinieerd (door #) wordt iedere regel
 door het invoegen van spaties op 72 posities gebracht.

0410
&
*
)'-'  'INDELINGS-MODE' UIT
)-----------------------
*
+Het teken '-' laat de 'document processor' weten, dat indien de
 'indelings-mode' aan is gezet met behulp van het teken '+'
 deze weer uitgezet moet worden. Alle regels die nu volgen worden
 geprint zoals ze op het beeldscherm zijn ingegeven.
```

```
0420
*
*
)'>'  ZET PAGINANUMMER
)---------------------
*
+Het teken '>' laat de 'document processor' weten, dat er op de volgende
 pagina een ander paginanummer moet komen als dat er berekend is.
-
 Voorbeelden:
*>15      (volgende pagina krijgt paginanummer 15)
*>1       (volgende pagina krijgt paginanummer 1)

0430
*
*
)'$'  ZET REGELS PER PAGINA
)--------------------------
*
+Het teken '$' laat de 'document processor' weten, hoeveel regels er op
 een pagina afgedrukt moeten worden.

0440
*
*
)'#'  ZET TEKENS PER REGEL
)-------------------------
*
+Het teken '#' laat de 'document processor' weten, hoeveel tekens er op een
 regel moeten worden afgedrukt. Dit stuurteken heeft echter alleen dan een
 funktie wanneer het in kombinatie met het teken '+' wordt gebruikt.

0450
&
*
)'~'  ZET AANTAL SPATIES BIJ NIEUWE ALINEA
)-----------------------------------------
*
+Het teken '~' laat de 'document processor' weten, hoeveel spaties er aan
 het begin van een alinea aan het begin van de eerste regel moeten worden
 afgedrukt.
```

```
0460
*
*
)')' .ZET IN HET MIDDEN VAN DE REGEL
)-----------------------------------
*
+Het teken ')' laat de 'document processor' weten, dat de achter dit teken
 aanwezige tekst in het midden van de regel moeten worden afgedrukt.
 Dit kan bijvoorbeeld gebruikt worden bij de hoofdstuknaam.
-
 Voorbeeld:
*)INHOUDSOPGAVE
 )=============

0470
&
*
)APPENDIX
)========
*
+Als uitgebreid voorbeeld van een met de 'system editor' opgebouwd
 en met de 'document processor' geprint tekstbestand, vindt u achter
 deze handleiding een print van deze beschrijving, inklusief alle
 stuurtekens.
*
-    Let u hierbij op de gebruikte technieken:

0480
*     1. Er mag geen tekst rechtstreeks achter het
         paragraafnummer worden ingegeven.
*     2. De eerste positie van elke regel wordt uitsluitend
         gebruikt voor stuurtekens. Dit is echter niet
         verplicht, maar verhoogt wel de leesbaarheid van
         het tekstbestand tijdens eventuele korrekties.
```